ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΑΡΟΥΣΙΑΣΗ / ΕΞΕΤΑΣΗ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Αθανασάκης Μιχάλης

Μεταπτυχιακός Φοιτητής

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Επόπτης Μεταπτ. Εργασίας: Καθηγητής Ε. Μαρκάτος

Παρασκευή, 22/1/2016, 11:00

Αίθουσα "Σ. Ορφανουδάκης, ΙΤΕ

"Παράβιαση αμυνών σε μεταγλωττιστές δυναμικής παραγωγής κώδικα σε περιηγητές διαδικτύου"

ΠΕΡΙΛΗΨΗ

Οι επιθέσεις με την χρήση επιστρεφόμενου προγραμματισμού (ROP) είναι η πιο διαδεδομένη μορφή επιθέσεων σε επίπεδο χρήση και λειτουργικών συστημάτων. Η δημιουργία αμυντικών τεχνικών έχει κάνει την επίθεση πιο δύσκολη. Ήδη οι επιτιθέμενοι έχουν αρχίσει να εκμεταλλεύονται μεταγλωττιστές δυναμικής παραγωγής κώδικα (JIT), διαθέσιμους σε όλους τους γνωστούς περιηγητές διαδικτύου, για την δημιουργία κακόβουλου πηγαίου κώδικα. Ο κακόβουλος κώδικας δημιουργείται με την μορφή πηγαίου κώδικα ή gadgets και μετά το τέλος της JIT μεταγλώττισης οι επιτιθέμενοι μπορούν να τον εκμεταλλευτούν.

Αναγνωρίζοντας την απειλή οι δημιουργοί περιηγητών έχουν αρχίσει να εισάγουν μηχανισμούς άμυνας στις JIT μηχανές μεταγλώττισης κώδικα. Σε αυτή τη διατριβή δείχνουμε ότι παρόλο που όλες αυτές οι άμυνες είναι διαθέσιμες σήμερα δεν είναι αρκετές για να προστατέψουν το λογισμικό από την δυναμική δημιουργία κακόβουλων κομματιών κώδικα (gadgets). Δείχνουμε ότι η δημιουργία gadgets είναι δυνατή σε δύο διαδεδομένους διαδικτυακούς περιηγητές. Παραβιάζουμε το Mozilla Firefox παρέχοντας του κακόβουλη JavaScript, η οποία μετά την μεταγλώττιση μας παρέχει όλα τα απαραίτητα gadgets για την

επίτευξη της επίθεσης. Επίσης εκμεταλλευόμαστε το Internet Explorer 64-bit ώστε να εκτελέσουμε κακόβουλο λογισμικό. Το Internet Explorer διαθέτει πολλούς μηχανισμούς άμυνας τελευταίας γενιάς, κάποιοι μη-καταγεγραμμένοι έως τώρα. Παρόλα αυτά δείχνουμε ότι δεν είναι αρκετοί για την εξασφάλιση της ασφάλειας του λογισμικού. Καταγράφουμε όλες τις αμυντικές τεχνικές με σκοπό να βοηθήσουμε άλλους ερευνητές.

Επιπροσθέτως, εκτός από την παρουσίαση τεχνικών για την δημιουργία ROP gadgets δείχνουμε δυνατούς τρόπους εντοπισμού τους με σύγχρονο τρόπο, κάνοντας τις άμυνες που έχουν δημιουργηθεί για αυτό το σκοπό μη αποδοτικές. Επίσης πραγματοποιούμε μια ανάλυση πάνω σε μια από τις πιο σημαντικές μορφές άμυνας, το constant blinding. Σκοπός του είναι να προστατεύει τις τιμές αριθμών άνω των 3-bytes μέσα στην μνήμη κάνοντας την δημιουργία gadget αδύνατη. Δείχνουμε ότι κάτι τέτοιο δεν είναι αποδοτικό και ότι η προστασία όλων των τιμών μπορεί να επιφέρει την δημιουργία μέχρι και 80% παραπάνω εντολών.

Athanasakis Michalis
M.Sc. Thesis
Computer Science Department
University of Crete
Master's Thesis Supervisor: Professor E. Markatos

Friday, 22/1/2016, 11:00
Room "S. Orphanoudakis", ITE

"Bypassing Defenses of Just-In-Time Compilers in Modern Browsers"

ABSTRACT

Return-oriented programming (ROP) has become the dominant form of vulnerability exploitation in both user and kernel space. Many defenses against ROP during run-time make it much harder. Attackers have already started exploiting Just-in-Time (JIT) engines, available in all modern browsers, to introduce their (shell) code (either native code or reusable gadgets) during JIT compilation, and then taking advantage of it.

Recognizing this immediate threat, browser vendors started employing defenses for hardening their JIT engines. In this thesis, we show that –no matter the employed defenses --JIT engines are still exploitable using solely dynamically generated gadgets. We demonstrate that dynamic ROP payload construction is possible in two modern web browsers without using any of

the available gadgets contained in the browser binary or linked libraries. First, we exploit an open source JIT engine (Mozilla Firefox) by feeding it malicious JavaScript, which once processed generates all required gadgets for running any shellcode successfully. Second, we exploit a proprietary JIT engine, the one in the 64-bit Microsoft Internet Explorer, which employs many *undocumented, specially crafted* defenses against JIT exploitation. We manage to bypass all of them and create the required gadgets for running any shellcode successfully. All defensive techniques are documented in this thesis to assist other researchers.

Furthermore, besides showing how to construct ROP gadgets on-the-fly, we also show how to discover them on-the-fly, rendering current randomization schemes ineffective. Finally, we perform an analysis of the most important defense currently employed, namely constant blinding, which shields all three-byte or larger immediate values in the JIT buffer for hindering the construction of ROP gadgets. Our analysis suggests that extending constant blinding to all immediate values (i.e., shielding 1-byte and 2-byte constants) dramatically decreases the JIT engine's performance, introducing up to 80% additional instructions.